Gyroscope calibration

Ali Baharev, PhD

TÁMOP 4.2.2 “Szenzorhálózat alapú adatgyűjtés
és információfeldolgozás” workshop
September 2-3, 2010
Szeged, Hungary

What we have - Shimmer Wireless Sensor Platform

Sensors

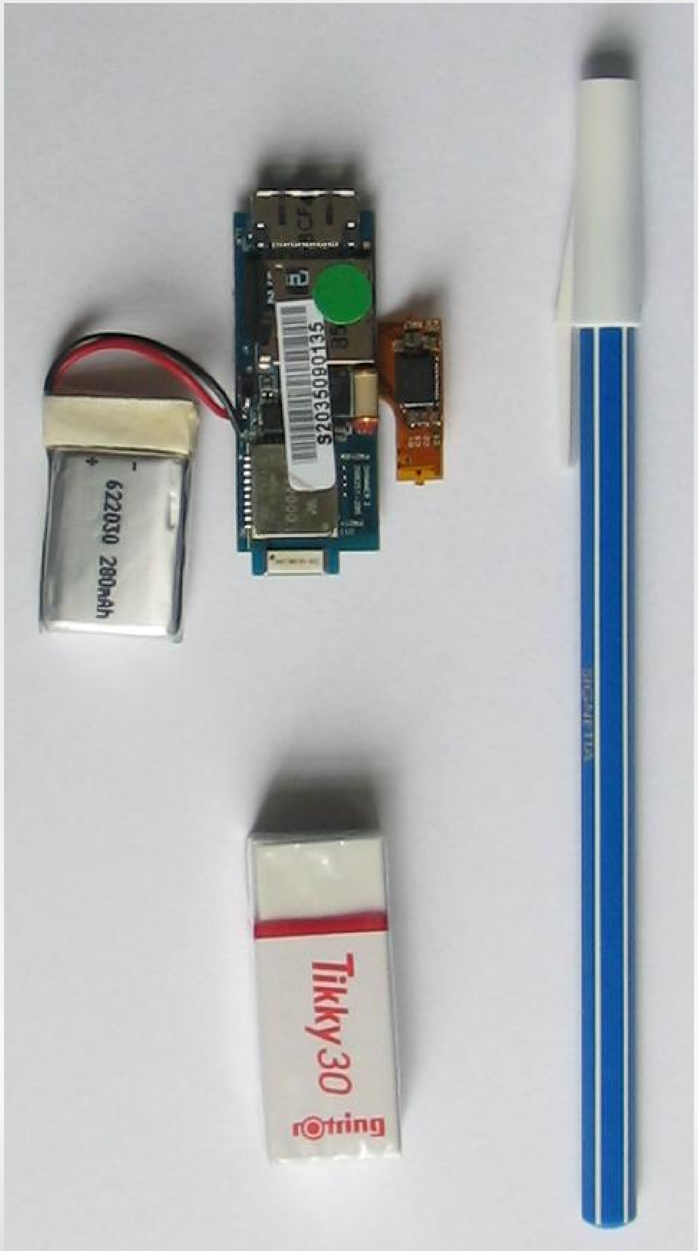
- 3D accelerometer, Freescale MMA7260Q ±1.5/2/4/6g (1g≈9.81m/s^2)
- 3D gyroscope; two integrated dual-axis angular rate gyroscopes InvenSense 500 series

Processing

- MSP430™16-bit Ultra-Low Power MCU @ 8 MHz
- 10Kbyte RAM, 48Kbyte ROM
- 8 Channels of 12bit A/D

Battery

- Integrated Li-ion, 280 mAh, 3.7 V



What we have - Shimmer Wireless Sensor Platform (continued)

Radios

- 2.4 GHz IEEE 802.15.4 Chipcon CC2420
- Mitsumi WML-C46N CSR based Class 2 Bluetooth Radio

Storage

- 2 GB Micro SD card

Form factor

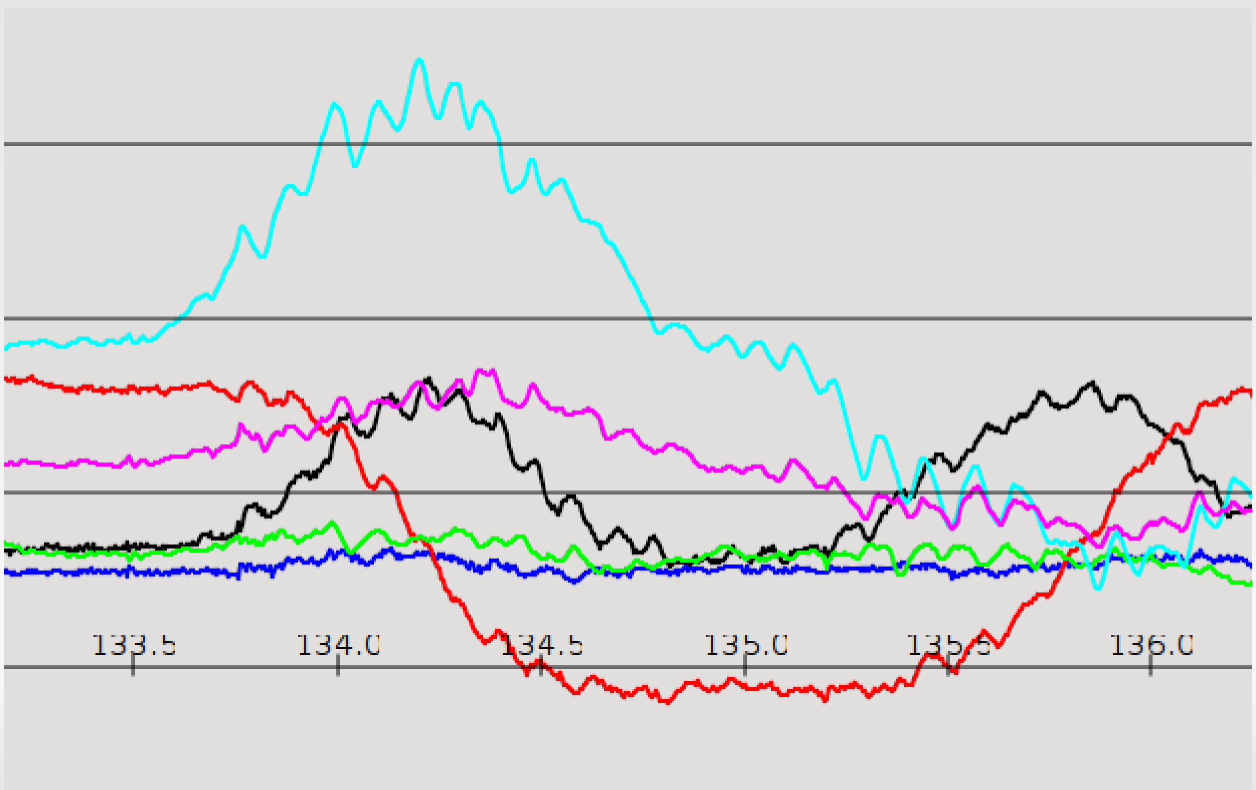
- Small form factor
- 50mm x 25mm x 12.5mm
- Light weight: 15 grams

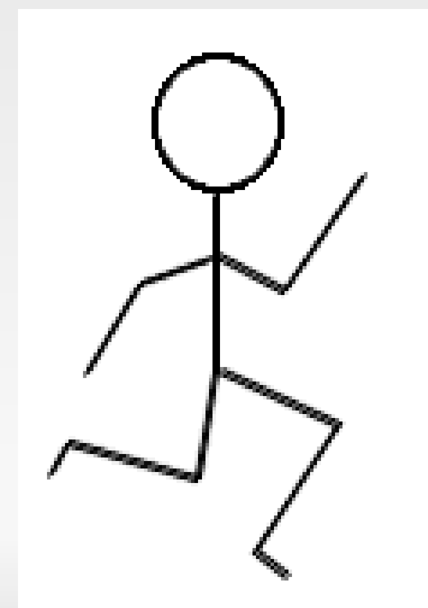
Software

- TinyOS event driven OS for WSN
- Open source

What we want - Gait analysis

- Analysis of measurable parameters of human gait
- From the output of the sensors –> reconstruct the orientation of the limbs in time
- The orientation, described by angles and position

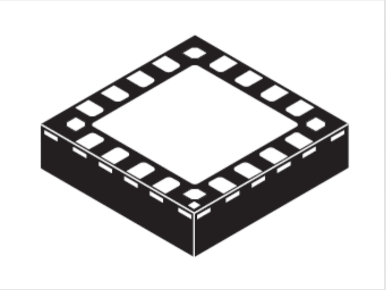


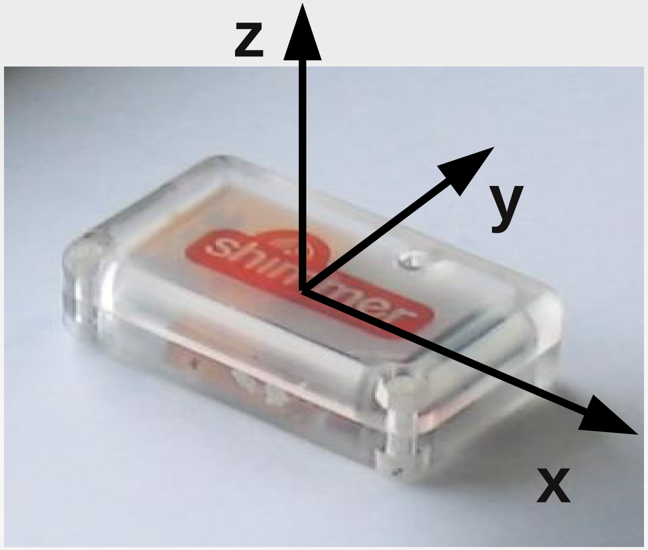


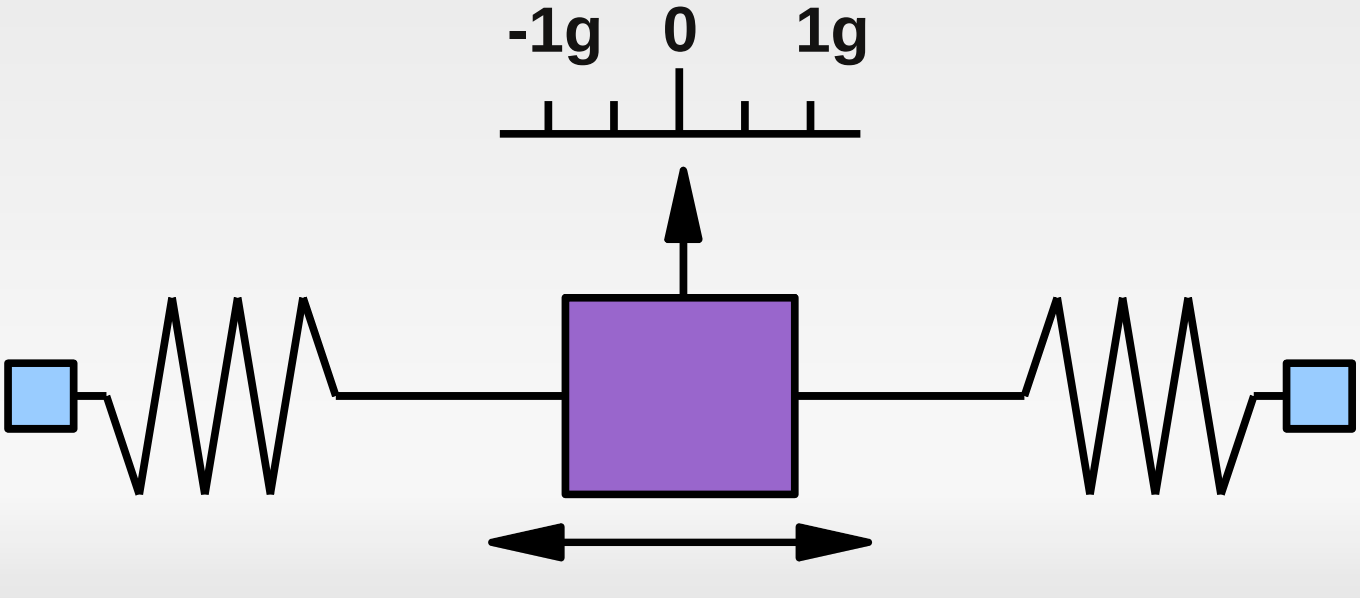
Sensors: accelerometer

Accelerometer

- 3 Axis Accelerometer, Freescale MMA7260Q
- Sensitivity: 800 mV/g @ 1.5g
- 12 bit analogue digital converter –> integer number
- Resolution: $(1.5g + 1.5g)/(2^{12}) \approx 7 \cdot 10^{-4}$ g/unit



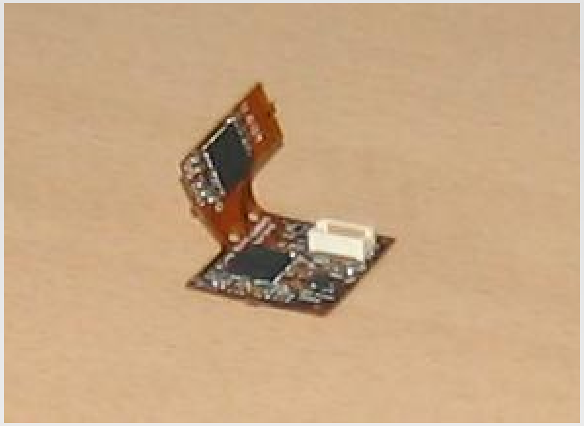




Sensors: gyroscopes

Gyroscope

- 2 integrated dual-axis, InvenSense 500 series
- Measures angular rate
- Full scale range: +/- 5000 deg/s
- Sensitivity: 2 mV/deg/s
- 12 bit ADC –> integer number



Output in the static case

- Offset (**not zero**), actual value depends mainly on chip, plus temperature, *etc.*
- Accelerometer: constant value corresponding to 1g (gravity of Earth)

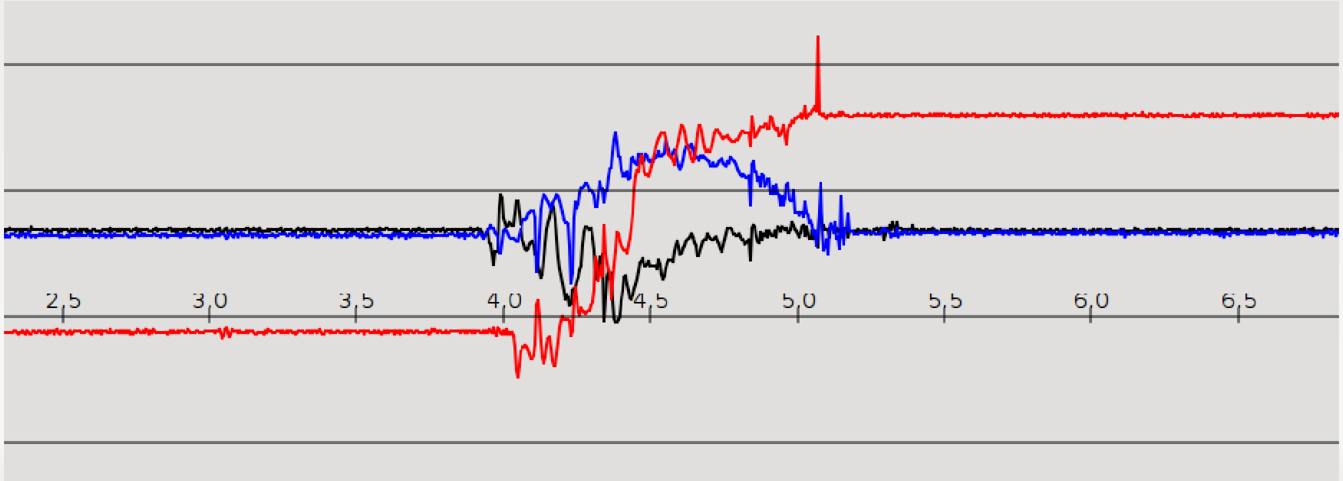
Static calibration of the accelerometer

Assumption: measured value is a linear function of the acceleration (linear transfer function)

Calibration: find the gain matrix (9 unknowns) and offset vector (3 unknowns)

acceleration [m/s^2] = gain·(measured value) − offset

Place the mote on each of its six side and record the output acceleration: (±1g, 0, 0); (0, ±1g, 0); (0, 0, ±1g)

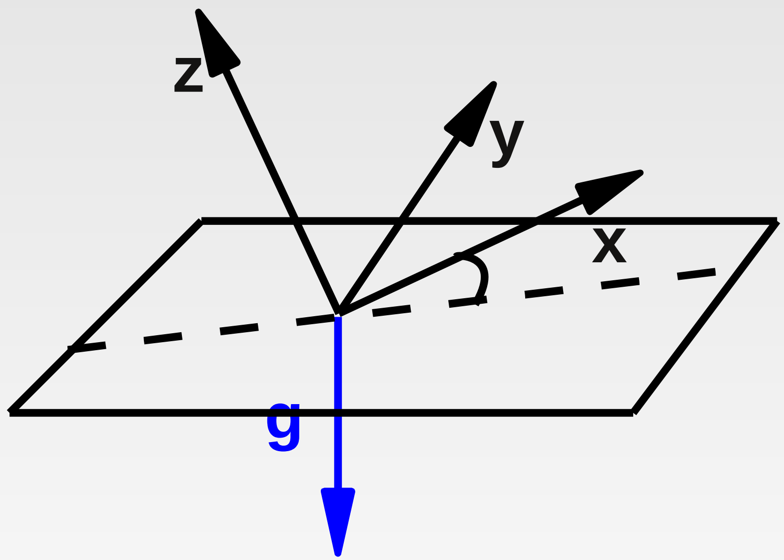


Gives an overdetermined system of linear equations (18 equations and 12 unknowns); linear least-squares, analytic solution (SVD)

What can we compute from acceleration?

In the static case constant 1g pointing downwards is measured

The angle between an axis of the device and the horizontal plane can be computed

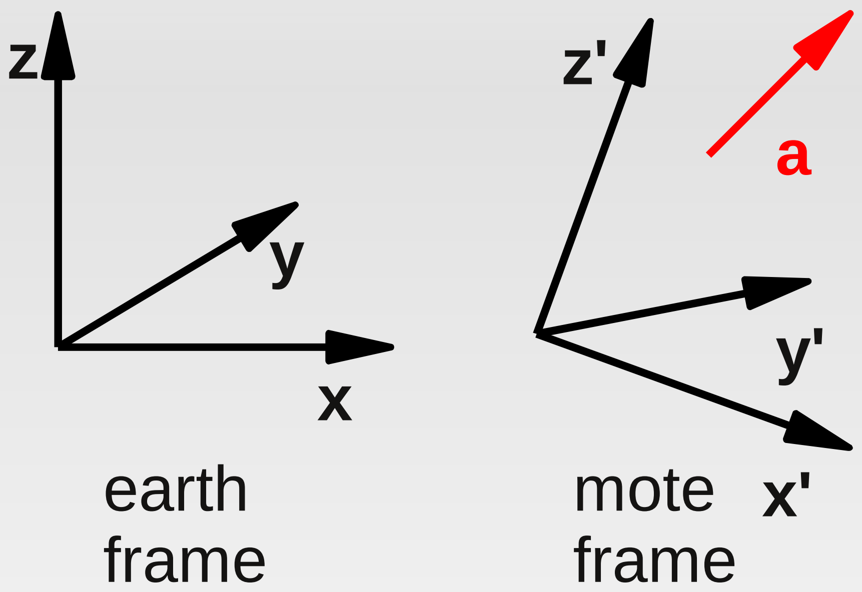


There is no way to compute orientation from the acceleration data, that would require additional data (for example where north is from a magnetometer)

Can we compute speed or position from a(t)?

$$v(t)=v(0)+\int_0^t (a(\tau)-g)\,d\tau$$

$$r(t)=r(0)+\int_0^t v(\tau)\,d\tau$$



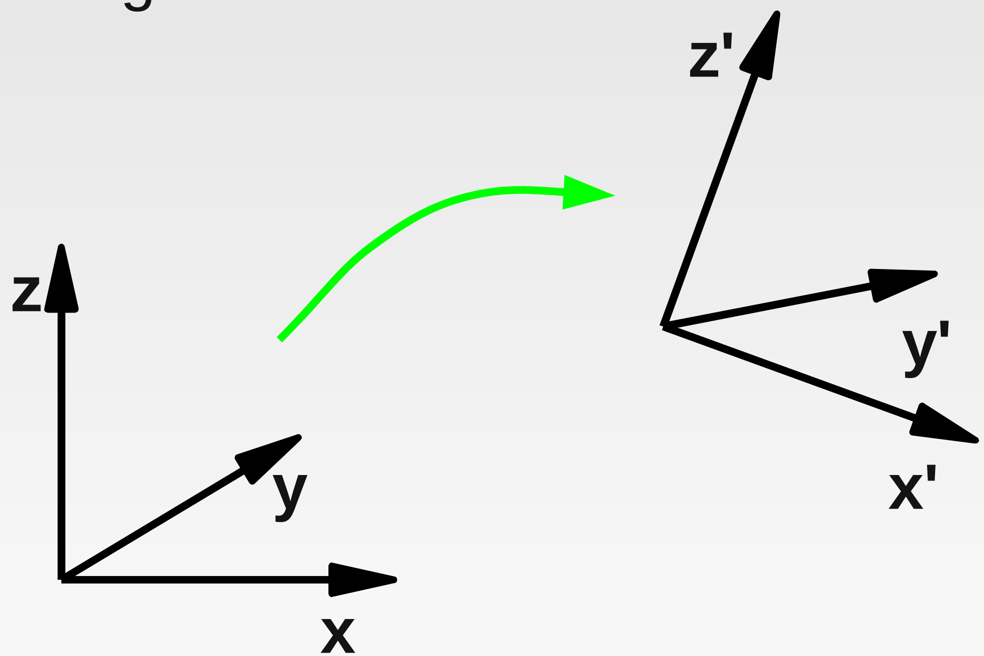
The a(t) vector is measured in the **mote frame** of reference but we would like to track the mote in the **earth frame**.

Transformation is needed from one frame to the other –> **rotation**

Rotation

We need rotation to transforms the acceleration vectors from the mote frame of reference to the earth reference

Rotation: linear transformation, preserves lengths of vectors and angles between vectors



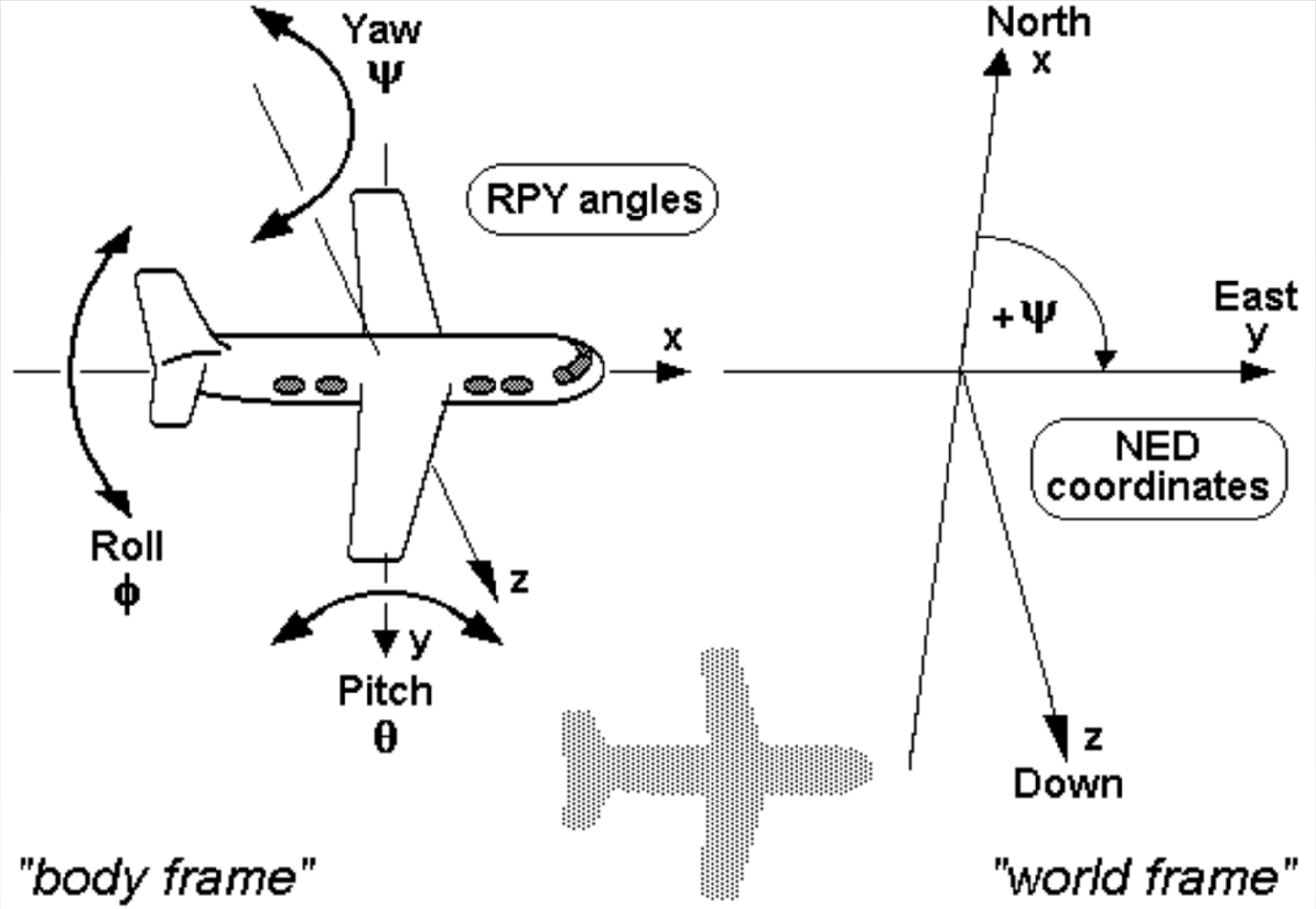
$$\begin{array}{c|ccc} & x & y & z \\ \hline x' & r_{xx'} & r_{yx'} & r_{zx'} \\ y' & r_{xy'} & r_{yy'} & r_{zy'} \\ z' & r_{xz'} & r_{yz'} & r_{zz'} \end{array}$$

Rotation matrix

Rotation (continued)

Rotation representations

- Rotation matrix
- Euler angles
- Angle/axis
- Quaternion



Rotation is uniquely defined by 3 angles

Performance comparisons of rotation methods

Storage requirements

matrix	9
quaternions	4
angle/axis	3

Performance comparison of rotation chaining operations

	multiplies	add/substr.	total
matrix	27	18	45
quaternions	16	12	28

Performance comparison of vector rotating operations

	multiplies	add/substr.	sin/cos	total
matrix	9	6	0	15
quaternions	21	18	0	39
angle/axis	23	16	2	45

Infinitesimal rotations

Rotation in 3D is generally not commutative (neither is matrix multiplication)

The order in which infinitesimal rotations are applied is irrelevant

Rotation matrix of infinitesimal rotations along the x, y, z axis:

$$\begin{bmatrix} 1 & -d\Theta_z & d\Theta_y \\ d\Theta_z & 1 & -d\Theta_x \\ -d\Theta_y & d\Theta_x & 1 \end{bmatrix}$$

Gives a recipe to update the rotation matrix from gyro signals

Updating the rotation matrix from gyro signals

W. Premerlani and P. Bizard; *Direction Cosine Matrix IMU: Theory*

$$R(t+dt) = R(t) \begin{bmatrix} 1 & -d\Theta_z & d\Theta_y \\ d\Theta_z & 1 & -d\Theta_x \\ -d\Theta_y & d\Theta_x & 1 \end{bmatrix}$$

$$d\Theta_x = \omega_x dt$$
$$d\Theta_y = \omega_y dt$$
$$d\Theta_z = \omega_z dt$$

Sources of errors

- Finite time step
- Quantization error: finite digital representation

The rotation matrix must be corrected –> **renormalization** at each point (no divisions or square roots)

Calibration of the gyroscopes

Calibration: find the gain matrix (9 unknowns) and
offset vector (3 unknowns; linear transfer function)

angular rate [rad/s] = gain·(measured value) − offset

Place the mote on each of its six side and record the output angular rate: (±45rpm, 0, 0); (0, ±45rpm, 0); (0, 0, ±45rpm)

A small error in the offset accumulates –> huge error in orientation over time, **drift**

Drift cancellation

Drift
The integrated effects over time of a slowly varying offset and noise.
The drift must be eliminated, requires an orientation reference vector that does not drift.

Online methods
Kalman Filter: operates on all the measured data points individually

Our approach
Offline, operates on the whole data set but manipulates only 12 variables (gain and offset of the gyro)

Assumption: the mote cannot accelerate for long in any direction otherwise it would hit the wall of the room

Drift cancellation off-line with regression

Assumption: the mote cannot accelerate for long in any direction otherwise it would hit the wall of the room. 'On average' the measured acceleration points downwards (gravitational acceleration).

Rotate the measured acceleration vectors so that 'on average' they point into the same direction (we do not know where downward is)

$$\max \left| \sum_{i=0}^{N} R_i a_i \right|$$

$$R(0) = I$$

$$R(i) = R(i-1) G(i-1) \quad \text{for} \quad i = 1 \ldots N$$

$$G(i): \quad \text{from gyro signals}$$

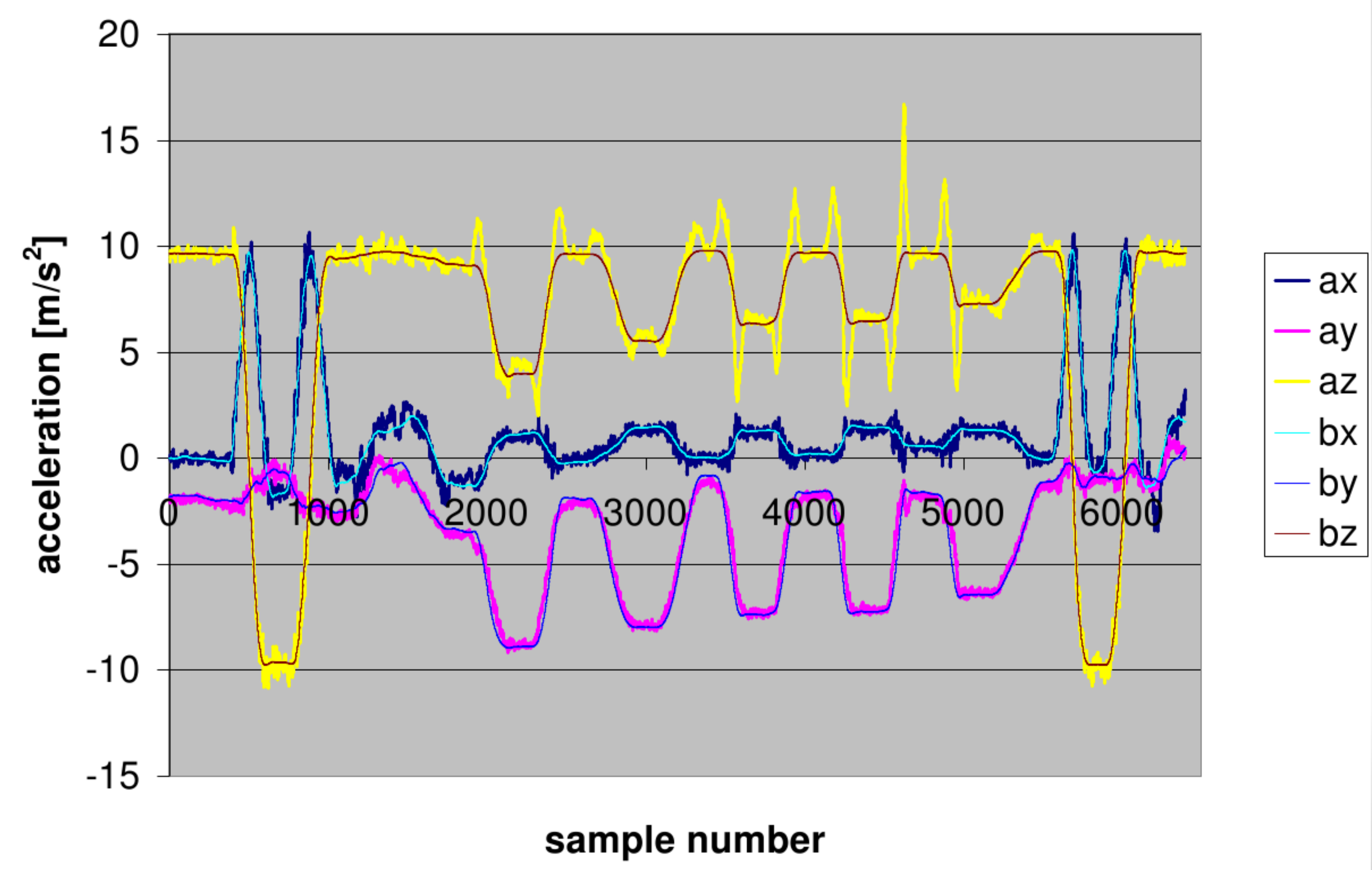
Variables: gyro gain and offset
Nonlinear programming problem

Software

- **NLP**, a nonlinear programming problem to be solved
- **IPOPT**, general purpose NLP solver (line search filter method) remarkably robust
- **C++ API** is used, only the objective has to be implemented
- **Automatic differentiation** (AD): the gradient is not approximated with numerical differentiation but automagically computed with AD (our own C++ library)
- **L-BFGS** (approximates the inverse Hessian matrix) to further speed up the computations

Results: acceleration

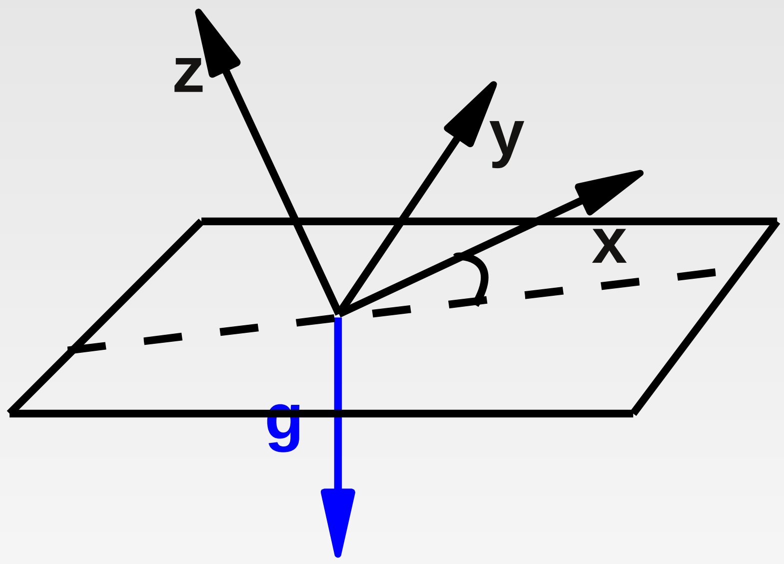
Comparing the measured acceleration (ax, ay, az) and the gravitational acceleration (bx, by, bz) in the mote frame



What can we compute from acceleration?

In the static case constant 1g pointing downwards is measured

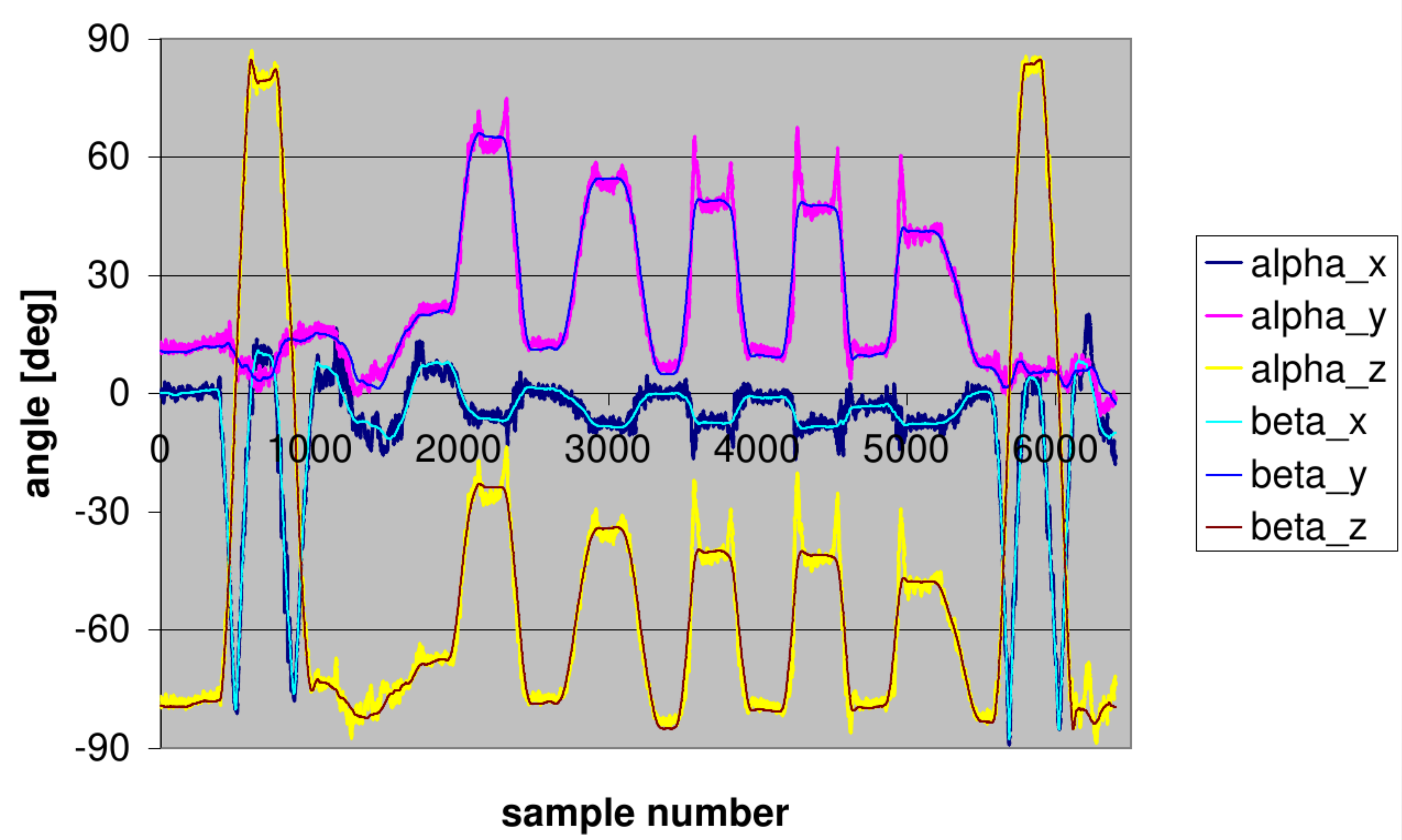
The angle between an axis of the device and the horizontal plane can be computed



There is no way to compute orientation from the acceleration data, that would require additional data (for example where north is from a magnetometer)

Results: angles

Cross-check: if the mote is static, the orientation computed from the measured acceleration ONLY (alpha) must coincide with that computed from the gyro signals ONLY (beta)

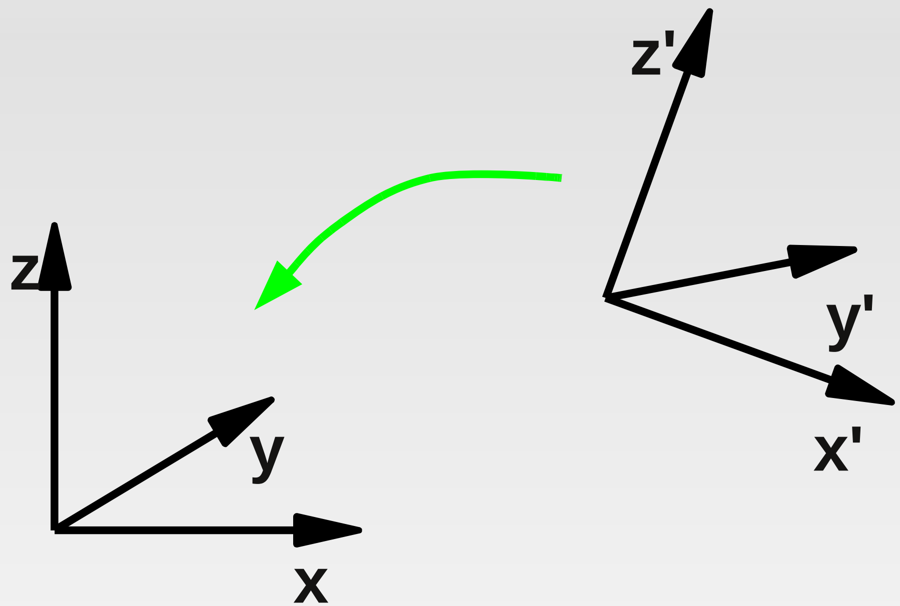


Future plans

Now, the measured values in the mote frame can be transformed to the earth frame

$$v(t)=v(0)+\int_0^t (a(\tau)-g)\,d\tau$$

$$r(t)=r(0)+\int_0^t v(\tau)\,d\tau$$



However we cannot integrate directly for the same reason as the gyro –> a similar drift cancellation procedure is required for the gain and offset of the *accelerometer*

Acknowledgements

Péter Ruzicska: Qt application providing the calibration modules

Miklós Tóth: Java application for processing files in binary format

Miklós Maróti: supervising the research

The presented work was supported by the Grant TÁMOP-4.2.2/08/1/2008-0008